

Oracle Label Security Best Practices

*An Oracle White Paper
June 2008*

Oracle Label Security Best Practices

- Introduction 3
- Installation Guidance 3
- Implementation Overview 4
- Oracle Label Security Administration 5
- Important Enforcement Exemptions..... 6
- Planning Your Data Labels 6
- Planning Your User Labels 10
- Planning your Enforcement Options..... 12
- Oracle Label Security Privileges 13
- Performing an Access Authorization Analysis..... 15
- Review and Document 15
- Labeling Legacy Data..... 15
- Oracle Label Security and Applications 16
- Performance Considerations 17

Oracle Label Security Best Practices

INTRODUCTION

Oracle Label Security mediates access based on data sensitivity labels, referred to in this document simply as *data labels* and user label authorizations, referred to in this document simply as *user labels*. Oracle Label Security provides multi-level security capabilities for government and defense applications. Oracle Label Security benefits commercial organizations attempting to address numerous access control challenges including those associated with database and application consolidation, privacy laws and regulatory compliance requirements. Available for Oracle8i Enterprise Edition databases and higher, Oracle Label Security has been evaluated to the international common criteria at EAL4. Most recently Oracle Label Security 10.2.0.3 completed an independent evaluation to the common criteria at EAL4+. Oracle Label Security 11g manageability is completely integrated with Oracle Enterprise Manager. This integration with Enterprise Manager replaces the Oracle Policy Manager tool that was available with previous releases of the Oracle Database.

INSTALLATION GUIDANCE

Please note that Oracle Label Security does not install by default with the Oracle Enterprise Edition. When running the Oracle installer you must choose the *Custom* installation option and manually check the box beside Oracle Label Security. Please note that you must run the Oracle Installer, as it is not sufficient to simply run the associated Oracle Label Security object creation catalog scripts. The installer will re-link the Oracle binary executable during the installation of Oracle Label Security. After running the Oracle Installer you should also run the Oracle Database Configuration Assistant (DBCA). DBCA will execute the necessary catalog scripts to create the Oracle Label Security administration account, tables, views, functions and procedures.

Please note that if you have already successfully installed Oracle Database Vault 11g then Oracle Label Security is already installed. However, the administration account for Oracle Label Security will be locked by default. Oracle Database Vault customers have a restricted use license of Oracle Label Security allowing it to be installed. Creating Oracle Label Security specific policies through Oracle Enterprise Manager or the Oracle Label Security API requires a separate Oracle Label Security license not provided by Oracle Database Vault.

IMPLEMENTATION OVERVIEW

Much like any sophisticated security product, planning your deployment of Oracle Label Security is very important and will help avoid any potential problems. The steps below provide a basic guideline for deploying Oracle Label Security. The implementation can be performed using Oracle Enterprise Manager or the Oracle Label Security API / command line interface. It may be useful to work with a sample demonstration table early on to get a firm understanding of how data labels mediate access control as well as the various enforcement options available in Oracle Label Security.

Oracle Label Security Implementation Steps
1. Perform the data analysis steps recommended in this paper.
2. Create the Oracle Label Security policy.
3. Define necessary data label components including levels, compartments and groups.
4. Provision user labels (Max, Min, Default)
5. If you plan to label data rows, create the data labels for the policy using the components (levels, compartments and groups) already defined.
6. Apply the policy to the application tables. Note that once applied, no data will be accessible unless special privileges have been granted to the user.
7. Update legacy data with appropriate data labels using the techniques described in this paper.

ORACLE LABEL SECURITY ADMINISTRATION

The primary administration account for Oracle Label Security is the user account *LBACSYS*. This account contains the data dictionary tables that store the Oracle Label Security policies, data labels, user labels, enforcement settings, and Label Security protected objects.

Oracle Label Security administration is performed using the Oracle Enterprise Manager Database Console and navigating to the *Server* tab. On the *Server* tab page you will find a section called *Security*. Here you will find a link for Oracle Label Security administration. Note that all administrative tasks related to Oracle Label Security can be performed using the available PL/SQL API. The PL/SQL API is fully documented in the Oracle Label Security administrator's guide.

Delegated administration is possible using Oracle Label Security. When an Oracle Label Security policy is created, a new database role *policyname_DBA*. In the following code, the user *LBACSYS* creates a policy called *HRSEC* and gives the *user* the *HRSEC_DBA* role and authorizations to manage policy label components and label authorizations.

```
CONNECT LBACSYS
```

```
EXECUTE SA_SYSDBA.CREATE_POLICY('HRSEC', 'HR_LABEL');
```

```
GRANT HRSEC_DBA TO admin;  
GRANT EXECUTE ON sa_components TO admin;  
GRANT EXECUTE ON sa_user_admin TO admin;  
GRANT EXECUTE ON sa_label_admin TO admin;  
GRANT EXECUTE ON sa_policy_admin TO admin;  
GRANT EXECUTE ON sa_audit_admin TO admin;
```

Once granted, an administrator can execute the package and create label components, user labels, data labels and administer policies. Note that when any of the above packages are called, the package will check to see if the administrator has been granted the *policyname_DBA* role corresponding to the one specified on the input line. Since multiple Label Security policies can exist in a single database, each package requires the *policy name* to be supplied as an input argument. Optionally, individual administrators could be granted *execute* on different packages, enabling separation-of-duty to be customized.

IMPORTANT ENFORCEMENT EXEMPTIONS

The following exceptions are important to understand when using either Oracle Label Security and/or virtual private database (VPD) policies.

Exception	Description
SYS Objects	VPD and Label Security policies cannot be applied to objects in SYS schema.
SYSDBA Role	Any user that connects with the AS SYSDBA role is exempt from VPD and Label Security policies.
DIRECT Path Export	Oracle VPD policies and Label Security policies are not enforced during DIRECT path export.
EXEMPT ACCESS POLICY Database Privilege	Any user granted the Oracle Database EXEMPT ACCESS POLICY privilege, directly or through a database role is exempt from both VPD and Label Security policies.

PLANNING YOUR DATA LABELS

The first and most important step in planning your Oracle Label Security deployment is determining your organization's data label requirements. This means determining what *Data Labels* or *Sensitivity* you require to protect your information. Determining your data label requirements generally means analyzing your application and identifying the tables that you plan to protect with Oracle Label Security. This is best accomplished with the assistance of an application administrator or developer who has knowledge of the application schema. In most cases, only a small percentage of the application tables will require an Oracle Label Security policy. Once the candidate tables have been identified, the data contained in the tables needs to be evaluated. The assistance of a data analyst or someone with understanding of the data may be required. It is recommended that application data that may be stored in the future be considered as well. This will create a robust set of initial label components.

Data Labels contain 3 components - a level, optional compartments and optional groups.

Data Label Component	Label Component Description
Levels	The <i>level</i> is a hierarchical component that denotes the sensitivity of the data. Each and every data label <i>must</i> have a level. A typical organization might define levels such as <i>Confidential</i> , <i>Sensitive</i> and <i>Highly Sensitive</i> .
Compartments	The <i>compartment</i> component is optional and is sometimes referred to as a category and is non hierarchical. Typically one or more compartments are defined to compartmentalize data. Compartments might be defined for a specific type of data, knowledge area or project that requires special approval.
Groups	The <i>group</i> component is optional and is very similar to a compartment with a few exceptions. Each group can have a parent child relationship. Groups are most often used to segregate data by organization.

Oracle Label Security provides flexibility enabling you to customize *Data Labels* to your specific requirements.

Sample Policy and Label Component Matrix	HR Policy	Law Enforcement Policy	Government Policy
Levels	Confidential Sensitive Highly Sensitive	Level 1 Level 2 Level 3	Confidential Secret Top Secret
Compartments	PII Data Investigation	Internal Affairs Drug Enforcement	Desert Storm Border Protection
Groups	HR	Local Jurisdiction FBI Dept of Justice	NATO Homeland Security

Note that a single Oracle Label Security policy can have up to 999 levels and up to 9999 compartments and groups. However, most organizations have fewer than 5 levels.

The external or text representation of a *Data Label* uses colons and commas to separate the various components. For example, the data label *Sensitive: Alpha, Beta : UK* contains the level *Sensitive*, two compartments *Alpha* and *Beta* and one group *UK*.

Internally, Oracle Label Security uses a numeric identifier called a *label tag* for each sensitivity label. Label tags are established when creating the *Data Label*. Label tags are stored with each row in a protected table. The label tags are stored in a column defined by the administrator when a policy is created. The administrator can choose to have the column appended to an application table as a *Hidden* column. Appending the column as a *Hidden* column will eliminate any possibility of existing *Update or Insert* statements failing due to the fact the statement didn't qualify the columns names in the statement. It is important to note that the Oracle Label Security policy column can pre-exist in an application table prior to applying an Oracle Label Security policy. To take advantage of this, the application table column type must be *number (10)*. This allows applications to be designed with an Oracle Label Security policy column built-in.

Note that while sensitivity labels and label tags can be created dynamically at run time, Oracle highly recommends defining all sensitivity labels and associated label tags prior to their being used to label data.

When deciding whether to use compartments, groups, or both, it is important to understand their differences.

Data Label Contains	Required User Authorization
Level	User must be authorized to the level or higher. For example, in order for a user to access data labeled “Sensitive”, the user must have been authorized to at least the “Sensitive” level. The number assigned to the level determines its ranking.
Compartment	User must be authorized to all compartments listed in the data label. For example, in order for a user to access data labeled “Sensitive: Alpha, Beta”, the user must have been authorized to at least the “Sensitive” level and to both the “Alpha” and “Beta” compartments. Unlike levels, the number assigned to a compartment has no meaning other than determining the display order of multiple compartments when using the <i>label_to_char</i> function or similar functions.
Group	User must be authorized to at least one of the groups listed in the data label or be authorized to a parent group. For example, in order for a user to access data labeled “Sensitive: Alpha, Beta : United States, Europe”, the user must have been authorized to at least the “Sensitive” level, to at least one of the groups “United States” or “Europe” and to both Compartments “Alpha” and “Beta”. Note the colon separating the level, compartment and group sections in the label. Unlike levels, the number assigned to a group has no meaning other than determining the display order of multiple groups when using the <i>label_to_char</i> function or similar functions.

If the application has an entity relationship (ER) diagram, it may be useful to annotate on the diagram the range of data labels for each entity.

PLANNING YOUR USER LABELS

Oracle Label Security user labels must be established by the security administrator before an application user can access an application table protected by Oracle Label Security. Note that when multiple policies are present in the database, separate user authorizations must be established for each policy.

For example, here are two tables *Data Sources* and *Customers*. Both tables have unique Label Security policies applied. In order for a user to be able to view data from both tables, the user would have to be assigned *User Labels* for both policies. In this case a user would have to have a *User Label* level equal to at least *Highly Sensitive* for *policy1* and equal to at least *Sensitive* for *policy 2*.

Data Sources Table

Source	Renewal Date	POL1_SECLAB
SW-R1	201001	Highly Sensitive
SW-R1	201002	Sensitive

Customers Table

Name	Location	POL2_SECLAB
ACME	New York	Sensitive
WIDGET	London	Confidential

In the unusual case where multiple policies are assigned to the same table, the policies are *'anded'* together. In this example, two policies, *policy1* and *policy2* are assigned to the same table. Each policy adds a hidden column to the base table. If a user has a *User Label* with a level of *Sensitive* and the group *Mergers* for *policy1* and has a *User Label* with a level of *Sensitive* for *policy2*, then the user would only see the project *Galaxy* because the first row also requires *Highly Sensitive* for *policy 2*.

Projects Table (2 policies applied)

Project Name	Location	POL1_SECLAB	POL2_SECLAB
Condor	New York	Sensitive: : Mergers	Highly Sensitive
Galaxy	HQ	Confidential	Sensitive

Oracle Label Security *User Labels* are comprised of multiple components. These components are defined as follows:

User Label Authorization	Authorization Description
Maximum Level	The maximum sensitivity level a user is authorized to access. For example this might be <i>Sensitive</i> or <i>Highly Sensitive</i> .
Minimum Level	The minimum sensitivity level a user is authorized to write data. For example, an administrator can prevent users from labeling data as <i>Confidential</i> by assigning a minimum level of <i>Sensitive</i> .
Default Level	The level used by default when a user connects to the database. For example, a user can set his or her default level to <i>Sensitive</i> . When he or she connects to the system, the default level will be initialized to <i>Sensitive</i> .
Row Level	The default level used to label data inserted into the database by the user through the application or directly through a tool such as SQL*Plus.
Read Compartments	The set of compartments assigned to the user and used during READ access mediation. For example, if a user has compartments <i>A, B and C</i> , he could view data which has compartments <i>A and B</i> but not data which has compartments <i>A, B, C and D</i> .
Write Compartments	The set of compartments assigned to the user and used during WRITE access mediation. For example, a user could be given READ and WRITE access to compartments <i>A and B</i> but READ-ONLY access to compartment <i>C</i> . If an application record was labeled with compartments <i>A, B and C</i> , the user would not be allowed to update the record because he or she does not have WRITE access on compartment <i>C</i> .
Read Groups	The set of groups assigned to the user and used during READ access mediation. For example, if a user had the group <i>Manager</i> , he could view data that has the <i>Manager</i> group but not data that had only the <i>Senior VP</i> group.
Write Groups	The set of groups assigned to the user and used during WRITE access mediation. For example, a user could be

	given READ and WRITE access to group <i>Senior VP</i> but READ-ONLY access to group <i>Manager</i> . If an application record was labeled with a single group, <i>Manager</i> , the user would not be allowed to update the record because he or she does not have WRITE access on the <i>Manager</i> group.
--	--

While it is possible to specify specific READ or WRITE permissions on individual compartments and groups, in most use cases users will have both READ and WRITE permissions on all compartments and groups they are authorized to access.

Please note that care should be taken to make sure that the total number of compartments and groups authorized to a specific user does not exceed a character string greater than 4000. When assigning the level, compartments and groups, each component is stored internally using 5 characters. For example, if a user is given access to the level “Sensitive” and the compartments “Alpha” and “Beta”, internally Label Security will use 15 characters. Since each user always has a single level, it is important to monitor the total number of compartments and groups the user is authorized to access. If a user should require access to all data, consider giving the user the READ authorization. Note that when a group is a parent group, the total number of child groups should be taken into consideration when below 4000 characters.

PLANNING YOUR ENFORCEMENT OPTIONS

Multiple Label Security policies can exist in the same database with different enforcement options. Policy enforcement options can be customized for each policy and for each protected table. When a Label Security policy is created, a default set of enforcement options can be established. When the policy is then applied to an individual table, the enforcement options can again be customized. For example, in some cases the *READ CONTROL* option may be sufficient because the database user is restricted from update and delete operations by the underlying database table privileges.

Policy Enforcement Option	Policy Enforcement Description
READ CONTROL	Applies policy enforcement to SELECT operations using the Oracle Label Security algorithm for read access.
INSERT CONTROL	Applies policy enforcement to INSERT operations using the Oracle Label Security algorithm for write access.
UPDATE CONTROL	Applies policy enforcement to UPDATE operations using the Oracle Label Security algorithm for write access.
DELETE	Applies policy enforcement to DELETE operations using the

CONTROL	Oracle Label Security algorithm for write access.
WRITE CONTROL	Applies policy enforcement on INSERT, UPDATE, and DELETE operations. If this option is set, it enforces INSERT_CONTROL, UPDATE_CONTROL, and DELETE_CONTROL.
LABEL DEFAULT	<p>If the user does not explicitly specify a label on INSERT, the user's default <i>row label</i> value is used. By default, the <i>row label</i> value is computed internally by Oracle Label Security using the user's label. The default value would be comprised of the default ROW LEVEL combined with the WRITE COMPARTMENTS and WRITE GROUPS.</p> <p>A user can set the row label independently, but only to:</p> <p>A level which is less than or equal to the level of the session label, and greater than or equal to the user's minimum level.</p> <p>Include a subset of the compartments and groups from the session label, for which the user is authorized to have write access.</p>
LABEL UPDATE	Applies policy enforcement to UPDATE operations that set or change the value of a label attached to a row. The WRITEUP, WRITEDOWN, and WRITEACROSS privileges are only enforced if the LABEL_UPDATE option is set.
LABEL CHECK	Applies READ_CONTROL policy enforcement to INSERT and UPDATE statements to assure that the new row label is read-accessible by the user after an INSERT or UPDATE statement.
NO CONTROL	Applies no enforcement options. A labeling function or a SQL predicate can nonetheless be applied.

ORACLE LABEL SECURITY PRIVILEGES

Oracle Label Security has several privileges that can be assigned to users and stored procedures. Examine privileged users and determine what if any privileges should be assigned.

PRIVILEGE	DESCRIPTION
READ	The READ authorization enforces no additional read access control. Access mediation is still enforced on UPDATE, INSERT and DELETE operations. Oracle Label Security makes no mediation check on SELECT operations.

FULL	<p>The FULL authorization turns off all Oracle Label Security access mediation. A user with the FULL authorization can perform SELECT, UPATE, INSERT and DELETE operations with no label authorizations. Note that Oracle SYSTEM and OBJECT authorizations are still enforced. For example, a user must still have SELECT on the application table. The FULL authorization turns off the access mediation check at the individual row level.</p>
WRITEDOWN	<p>The WRITEDOWN authorization allows a user to modify the level component of a label and lower the sensitivity of the label. For example, application data which is labeled <i>Highly Sensitive: Alpha, Beta</i> could be changed to <i>Sensitive: Alpha, Beta</i>. This authorization is only applicable to policies that use the label update enforcement option.</p>
WRITEUP	<p>The WRITEUP authorization allows a user to modify the level component of a label and raise the sensitivity of the label. For example, application data which is labeled <i>Sensitive: Alpha, Beta</i> could be changed to <i>Highly Sensitive: Alpha, Beta</i>. Note that the <i>Maximum Level</i> label authorization assigned to the user would limit modification. This authorization is only applicable to policies that use the label update enforcement option.</p>
WRITEACROSS	<p>The WRITEACROSS authorization allows a user to modify the compartments and groups in a label to any valid compartment and group defined in Oracle Label Security for the policy. For example, if data labeled <i>Sensitive: Alpha</i> could be modified to <i>Sensitive: Alpha, Beta</i> even though the user was not authorized for the <i>Delta</i> compartment. This authorization is only applicable to policies that use the label update enforcement option.</p>
PROFILEACCESS	<p>The PROFILE ACCESS authorization allows a user to assume the Oracle Label Security authorizations of another user. For example, user <i>Scott</i> who has access to compartments <i>A, B, and C</i> could assume the profile of user <i>Joe</i> who has access to compartments <i>A, B, C and D</i>. This functionality might be useful in an environment where an application uses a single application account for all application users. Note that the PROFILEACCESS privilege cannot be granted to a stored procedure.</p>

PERFORMING AN ACCESS AUTHORIZATION ANALYSIS

This step requires understanding the various roles and responsibilities of the user population. For example, a user might be designated an analyst, highly privileged user, or administrative user. Understanding the various roles and responsibilities may require the assistance of managers and security administrators. After the user population has been separated into one or more roles or functional areas, a comparison needs to be performed between the data labels and the user label requirements. These need to correspond correctly for each of the tables identified earlier. The reason this step is important is to prevent data from being assigned a sensitivity label that no user has access to. In other words, the information required to perform a specific job responsibility might be out of reach to the application user due to his or her user label. In the worse case, data might be assigned a data label that no user can access, effectively hiding the data.

Table	<div style="display: inline-block; border: 1px solid black; padding: 2px;"> User Data </div>	C	S	S:A:US	S:A,B:US,UK
	Assets	C::UK	No Access	No Access	No Access
C::US		No Access	No Access	Access	Access
Projects	C	Access	Access	Access	Access
	S	No Access	Access	Access	Access
	S:A:US	No Access	No Access	Access	Access
	S:B:UK	No Access	No Access	No Access	Access
	S:A,B:US	No Access	No Access	No Access	Access

REVIEW AND DOCUMENT

Review and document the information gathered. Include such information as a list of application tables that need to be protected, the reason why, as well as a list of label components and their meaning. This information will also be useful for applying other security controls as well such as Oracle Database Vault Realms, Data Masking and Tablespace Encryption. This document should become part of the enterprise security policy and should be considered sensitive and kept in a safe location.

LABELING LEGACY DATA

Once an Oracle Label Security policy is applied to an application table with READ CONTROL no rows will be visible until valid data labels have been assigned to each data row. This is because the label tag field will be NULL. Optionally you

can grant the administrator responsible for labeling the initial data the Label Security authorization *FULL*. This will allow the administrator to see all rows regardless of the data label and ensure that all legacy data rows are property labeled.

Several methods exist for labeling legacy data. The first method for labeling legacy data simply uses an update statement against the base table.

```
UPDATE SALES SET SECLAB =  
char_to_label:('HRSEC','S')  
WHERE REGION_ID = 104;
```

This statement updates the SALES table and sets the policy label column SECLAB equal to the internal label tag defined for SENSITIVE in the FINANCE policy and SALES column REGION_ID is equal to 104.

The second method for labeling legacy data is to switch database connections during the data load. If the policy applied to the SALES table includes the LABEL_DEFAULT option, the users default ROWLABEL value will be used to set initialize the label tag column.

```
CONNECT US_SALES_MGR  
INSERT INTO SALES (Col1, Col2, Col3) VALUES  
('ACME', .....);  
  
CONNECT EU_SALES_MGR  
INSERT INTO SALES (Col1, Col2, Col3) VALUES  
('WIDGET', .....);
```

The third method is to write a labeling function using PL/SQL. Oracle Label Security label functions are written in PL/SQL. An example of a labeling function can be found in the Oracle Label Security administrator's guide.

ORACLE LABEL SECURITY AND APPLICATIONS

Oracle Label Security supports common application architectures including situations where the middle-tier connects to the database using a single database account. To accomplish this, Oracle Label Security provides the ability for an authorized user to assume the label authorization profile of another user. The PROFILE_ACCESS authorization is required to execute the SET_ACCESS_PROFILE procedure.

Oracle Label Security does not enforce a mapping between a physical database account and the user name specified when establishing *User Labels*.

Applications can use one of the many Oracle SYS_CONTEXT variables in combination with the SET_ACCESS_PROFILE command. Applications using

Oracle Enterprise User Security can pass the `EXTERNAL_NAME` `SYS_CONTEXT` value to the `SET_ACCESS_PROFILE` command.

```
SQL> execute sa_session.set_access_profile
       ('PRIVACY',SYS_CONTEXT('userenv','EXTERNAL_NAME'));
```

Applications can also pass the `PROXY_USER` or `CLIENT_IDENTIFIER` as follows:

```
SQL> execute sa_session.set_access_profile
       ('PRIVACY',SYS_CONTEXT('userenv','PROXY_USER'));
```

```
SQL> execute sa_session.set_access_profile
       ('PRIVACY',SYS_CONTEXT('userenv','CLIENT_IDENTIFIER'));
```

PERFORMANCE CONSIDERATIONS

Performance is important to all applications. Adding new functionality to existing applications requires due diligence up front to minimize the performance impact. Oracle Label Security provides row level security, basically turning on a security check at each row prior to allowing access. The performance overhead will depend on a variety of factors including:

1. Number and size of tables protected by Label Security
2. Label Security enforcement options used

Identifying the tables that require a Label Security policy is an important part of the upfront analysis. If all rows in a table are always accessed, applying a Label Security policy that assigns a *data label* to each row is not recommended and is probably redundant. Careful consideration of where to apply Label Security will result in an efficient use of the technology. In some cases, other Oracle Database security features may be more appropriate for addressing a given requirement than assigning a *data label* to each row.

Carefully consider the enforcement options you apply to an application table and use only those that are necessary to meet your security requirements. Each additional security check performed by Oracle Label Security will add additional performance overhead.

Oracle also recommends defining the associated label tags so that they fall within the range associated with the level of the data label. For example, suppose the levels confidential and sensitive have been defined along with two compartments, alpha and beta. The number associated with Confidential is 5000 and the number associated with Sensitive is 10000. When the valid data labels are defined the associated label tags associated with confidential should be between 5000 and

10000. For example the data label *confidential: alpha* might have a label tag of 5050 and the data label *sensitive: alpha, beta* might have a label tag of 10055.

Existing composite indexes can be modified to include the policy column added by Label Security. This can substantially improve performance for complex queries.

Should any user or stored procedure need access to all data it is recommended that the user or stored procedure be given the Oracle Label Security specific privilege READ or FULL. This will help reduce overhead and increase performance.

Carefully consider the enforcement options you apply to an application table and use only those that are absolutely necessary to meet your security requirements. Each additional security check performed by Oracle Label Security will impact performance.

When labeling new data, Label Security label functions will have the most performance overhead as they will invoke an internal database trigger. Using the *LABEL DEFAULT* enforcement policy enforcement option will have the least performance overhead.

Depending on the application usage, consideration should be given to creating bitmap indexes on the column added by Oracle Label Security to the application table. The percentage of unique labels compared to the number of data rows is usually low. Bitmap indexes will slow down data loads but increase performance on *select* statements.

In warehouse environments Oracle partitioning can be used to with Oracle Label Security. This will provide query optimization through partition elimination. This is particularly useful for large tables. Label Security will quickly skip data that resides in partitions outside of the users label.

The example below would place all data with *Label Tags* less than 2000 in partition *sx1* and all data with *Label Tags* less than 3000 in partition *sx2* and all data with *Label Tags* less than 4000 in partition *sx3*. Partitioning based on the *Data Label* also physically separates data at the storage level based on its *Sensitivity*.

```
CREATE TABLE EMPLOYEE
  EMPNO NUMBER(10) CONSTRAINT PK_EMPLOYEE PRIMARY KEY,
  ENAME VARCHAR2(10),
  JOB VARCHAR2(9),
  SEC_LABEL NUMBER(10)
  TABLESPACE PERF_DATA
  PARTITION BY RANGE (SEC_LABEL)
  (partition sx1 VALUES LESS THAN (2000) NOLOGGING,
  partition sx2 VALUES LESS THAN (3000),
  partition sx3 VALUES LESS THAN (4000) );
```



Oracle Label Security Best Practice

June 2008

Author: Paul Needham

Contributing Authors: Peter Wahl

Oracle Corporation

World Headquarters

500 Oracle Parkway

Redwood Shores, CA 94065

U.S.A.

Worldwide Inquiries:

Phone: +1.650.506.7000

Fax: +1.650.506.7200

oracle.com

Copyright © 2008, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.